COMP 734 DISTRIBUTED SYSTEMS

Instructor: Prasun Dewan (FB 150, dewan@unc.edu)
Course Overview

This course will provide an implementation-oriented study of distributed systems. Some of the topics covered will include inter-process communication, group communication, synchronization, remote procedure call, peer to peer and centralized sessions, fire-walls, causal broadcast, atomic broadcast, scalability, fault tolerance, replication, and transactions/concurrency control. These are foundational concepts, which are becoming particularly relevant with the emerging areas of cloud computing and distributed games. These concepts will be introduced as layers in a general distributed infrastructure. Your projects will implement new layers and provide alternative implementations of some of the existing layers. When implementing a layer, you will act both as an application programmer, using abstractions of the layers below, and a systems programmer, defining abstractions for the layers above. The number of lines of code required by each layer will be relatively small; however the compositions of these layers will be complex.

The main difference between this course and a distributed programming/theory course is that it will address the design and implementation
## Downloads

<table>
<thead>
<tr>
<th>Download Description</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beau Halloween Simulation (Eclipse Project, <strong>uncompress</strong> and link to oeall17.jar or oeall22.jar)</td>
<td>beau_project.zip</td>
</tr>
<tr>
<td>Coupled Halloween Simulations (Eclipse project, <strong>uncompress</strong> and link to Beaj project and objecteditor)</td>
<td>CoupledTrickOrTreat.zip</td>
</tr>
<tr>
<td><strong>ObjectEditor</strong></td>
<td>oeall17.jar</td>
</tr>
<tr>
<td>Latest ObjectEditor (jar)</td>
<td>oeall22</td>
</tr>
<tr>
<td>GIPC (Github project library project)</td>
<td>GIPC</td>
</tr>
</tbody>
</table>
# Grade Distribution

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exams (Two midterms, no final)</td>
<td>40%</td>
</tr>
<tr>
<td>Assignments (Home work)</td>
<td>50%</td>
</tr>
<tr>
<td>Class work (10%)</td>
<td>10%</td>
</tr>
<tr>
<td>Fudge Factor (Class participation, other factors)</td>
<td>10%</td>
</tr>
</tbody>
</table>
GETTING HELP

Can discuss solutions with each other at a high level

Not at the code level

Sharing of code is honor code violation

Can help each other with debugging as long as it does not lead to code sharing

Assignments may contain solution in English (read only if stuck)
Getting Help and Class Discussion

We will be using Piazza for class discussion and getting help. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you do not get a response within a day or two on Piazza, please send mail.

Before posing a question, please check if this question has been asked before. This will reduce post clutter and reduce our burden. Repeat questions will be ignored by the instructors.

Piazza allows anyone to respond. So if you see a question that you think you can respond to, please do so, as that will reduce my burden and help you "teach" your fellow students.

This will be a form of class participation that will be noted when I allocate my fudge points!

Hope it works well

If you have any problems or feedback for the developers, email team@piazza.com.

Find our class page at: https://piazza.com/unc/fall2015/comp734/home
HALLOWEEN SIMULATION

Make Beau Anderson’s 401 Halloween implementation distributed
USE NON BLOCKING I/O

Distributed Non Blocking Simulation

Existing non distributed simulation

Java NIO
**Use Synchronous RMI**

- Distributed RMI-based Simulation
- Existing non-distributed simulation
- Java Object Serialization
- Sockets
- RMI
ASYNCHRONOUS GIPC

- Distributed RMI-based Simulation
- Existing non-distributed simulation
- Java Object Serialization
- Java NIO
Custom Serialization

- Distributed RMI-based Simulation
- Existing non distributed simulation
- GIPC
- Custom Serialization (a la Web Services)
- Java NIO
# Atomic Broadcast and Fault Tolerance

## Distributed RMI-based Simulation

## Fault Tolerant Atomic Broadcast

## Existing non-distributed simulation

## GIPC

## Custom Serialization (a la Web Services)

## Java NIO
INVERTED CLASS

Live Atomic Fault Tolerant Lectures

Slides and Youtube videos from Fall 13 offering

Class will involve quizzes, discussion and project work
# Lectures and Assignments

<table>
<thead>
<tr>
<th>Unit (Start Date)</th>
<th>Slides</th>
<th>Chapters</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>PPT PDF</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>YouTube</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Java Byte Communication</td>
<td>PPT PDF</td>
<td></td>
<td>Distributed Non-Blocking Halloween Simulation</td>
</tr>
<tr>
<td></td>
<td>YouTube-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>YouTube-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>YouTube-3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Current assignment is on the web - start working ASAP on it.